Technical Report
NREL/TP-XXXXX
November 2012

WT_Perf User Guide for Version 3.05.00

A. D. Platt and M. L. Buhl Jr.

Revised November 9, 2012
for WT_Perf v3.05.00a-adp

DRAFT

NREL IS A NATIONAL LABORATORY OF THE U.S. DEPARTMENT OF ENERGY, OFFICE OF ENERGY
EFFICIENCY & RENEWABLE ENERGY, OPERATED BY THE ALLIANCE FOR SUSTAINABLE ENERGY, LLC.
CONTRACT NO. DE-AC36-08GO28308

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

Available electronically at http://www.osti.gov/bridge
Available for a processing fee to U.S. Department of Energy
and its contractors, in paper, from:
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone: 865.576.8401
fax: 865.576.5728
email: mailto:reports@adonis.osti.gov

Available for sale to the public, in paper, from:
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
phone: 800.553.6847
fax: 703.605.6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/help/ordermethods.aspx

Cover Photos: (left to right) PIX 16416, PIX 17423, PIX 16560, PIX 17613, PIX 17436, PIX 17721

♻Printed on paper containing at least 50% wastepaper, including 10% post consumer waste.

DRAFT

DRAFT

Introduction

WT_Perf uses blade-element momentum (BEM) theory to predict the performance of wind turbines. It is a descendent of the PROP code originally developed by Oregon State University decades ago. Over the intervening years, many people from many organizations put their marks on the code and it has had several names. More recently, the staff at the NWTC rewrote the program, modernized it, added new functionality and algorithms, and partially rewritten it again.

Retrieving Files from the Archive

Create a folder for your WT_Perf files. You can then download the WT_Perf archive from our server at http://wind.nrel.gov/designcodes/simulators/wtperf. The file is named something such as "WT_Perf v3.05.00a-adp.exe," depending upon the version number. Create a WT_Perf folder somewhere on your file system and put this file there. You can double-click on it from Windows Explorer or by entering the file name (currently "WT_Perf v3.05.00a-adp.exe") at a command prompt, with the WT_Perf folder as the current directory. This will create some files and folders. Non windows users can download the tar.gz file (currently "WT_Perf v3.05.00a-adp.tar.gz").

Distributed Files

The files in the WT_Perf archive include the following:

AlphaChangeLog.txt	The list of changes to WT_Perf for the various alpha versions.
ArcFiles.txt	The list of files that are written to the archive.
Archive.bat	The batch file that creates the archive.
ChangeLog.txt	The list of changes to WT_Perf.
WT_Perf.exe	The WT_Perf executable file.
WT_Perf.pdf	This user's guide in PDF format.
CertTest*.*	Sample input, output, and verification files.
Source*.*	The source-code files for WT_Perf.

Using WT_Perf

WT_Perf must be run from the operating system's command prompt. Windows Vista/7 users can access the command prompt by right clicking on the folder while holding the shift key, then selecting *open command window here*. In older version of Windows the user will need to click Start, and then click Run and type "cmd" in the popup window. Once in the command prompt, navigate to the appropriate folder. The syntax for WT_Perf is then:

```
 WT_Perf <input file>
```

If you do not enter the input-file argument, WT_Perf will display the syntax to remind you. All output files use the same root name as the input file, but will have different extensions. The output file extensions are as follows:

.bed – the blade-element data
.ech – the echo of the input data
.oup – the primary output file

Linux and MAC users can build the program from the source code contained in the tar.gz file on the website since we do not distribute executable binaries for those. The source for the latest version of the NWTC_Library is also required (currently version 1.05.00). There is also a small makefile contained in the tar.gz file (the path names will need to be modified accordingly).

Creating the WT_Perf Input File

To create an input file, copy and edit one of the example *.wtp files from the CertTest folder. *Do not add or remove any lines*, except for the variable-length tables, such as the blade layout, the list of airfoil file names, or the list of combined cases. Do not depend on the values found in the sample input files to be accurate representations of the real turbines — many were modified for convenience.

A section-by-section description of an input file follows. variable names use the italicized *Letter Gothic* typeface. The line numbers on which the variables are found are indicated in square brackets such as [14]. Line numbers with a plus sign (+) after them require adjusting due to the number of specified segments (*NumSeg*) and airfoil tables (*NumAF*) within the input file.

Header

[1] The first line of the file states the type of file. You may change the line, but do not remove it or add additional lines.

Job Title

[2-3] You have two lines to describe the turbine model. WT_Perf copies the first of the two lines in the header to the output files.

Input Configuration

[5] *Echo* – This flag tells WT_Perf whether or not to echo the input data to the file "echo.out." If you set it to true, WT_Perf will write out the input values next to their descriptions. If WT_Perf crashes as a results of an input error, checking this file will help you figure out what caused the crash.

[6] *DimenInp* – If set to true, this flag tells WT_Perf to expect dimensional input parameters. If set to false, some parameters, such as the chord, are assumed to be nondimensional. If you want to use nondimensional input, divide parameters, such as the chord, by the rotor radius. The input-file comments tell you which parameters can be nondimensional. If a parameter can be normalized by the rotor radius, the comment states this in the units for the parameter. For example, the units for the hub radius are "[length or div by radius]," where the word "length" means

the parameter has units of length if dimensional, or it is divided by the radius of the rotor if normalized.

[7] *Metric* – This flag tells whether or not English units are used or if metric units are used. This parameter does not apply to the wind speed. The setting for wind speed is not affected by the *Metric* flag. Units used are as follows:

Measurement	**Metric**	**English**
time	seconds	seconds
length	meters	feet
mass	kilogram	slugs
force	newtons	pounds
angle	degrees	degrees

Model Configuration

This alpha version of WT_Perf uses a significantly different iteration algorithm to compute the induction factors than previous versions. For more information on how the new induction algorithm works in this version of WT_Perf, please see the paper by Maniaci.

[9] *NumSect* – The number of pie-wedge sectors around the rotor disk that are used in the calculations. If you set the *Tilt*, *Yaw*, and *ShearExp* (wind-shear exponent) to zero, you need only one sector. This is true because all calculations in all sectors are the same. WT_Perf ignores this parameter in this situation and analyzes one sector. If any of those three parameters are not zero, WT_Perf will use a minimum of four sectors in the analysis. There is virtually no upper limit for the number of sectors.

[10] *MaxIter* – Limits the number of iterations the Newton Raphson will perform while searching for the solution to the axial induction equations in the BEM routine. In most cases, if the Newton Raphson method can find the solution, it will do so within 15 iterations (usually in much fewer). If this method does not converge on a solution (there are situations where it will get stuck in a loop), computationally more expensive methods will then be tried. In general, the larger the *MaxIter* value is, the faster WT_Perf will run because the Newton Raphson method is more likely to converge.

[11] *NSplit* – Limits the number of iterations the binary search routine (also known as a half-interval search or bisection algorithm), BinSearch, may use while searching for the solution to the axial induction in the BEM routine. Ideally this number should be on the order of about 25. This routine is run only if the Newton Raphson routine fails to return a solution. It may also not yield solution or may step over a solution without finding it.

[12] *ATol* – This tells WT_Perf how little you want the BEM induction factors to change from one iteration to the next in order to consider it converged.

[13] *SWTol* – After BEM convergence for each element, WT_Perf uses the average induction for the entire rotor to compute the skewed wake correction, which it then applies to each element. WT_Perf will recompute the induction for each element using the newly corrected induction values for the initial estimates. It then uses the resulting average induction factor to compute and add the skewed-wake correction to each of those elements. WT_Perf continues this outer iteration until the correction changes by less than *SWTol*.

Algorithm Configuration

[15-16] *TipLoss* and *HubLoss* – These flags tell the code to turn on the Prandtl tip- and hub-loss algorithms. This is usually enabled for non-research work.

[17] *SWIRL* – This flag tells the code to enable the algorithms for the calculation of the tangential induction factor (swirl). This is generally set to true.

[18] *SkewWake* – This tells the code to correct the induction factor for a skewed wake. It is ignored if the *Tilt* and *Yaw* are both zero.

[19] *IndType* – Setting this flag to false eliminates the effects of induction from the algorithm. *IndType* should almost always be set to true.

[20] *AIDrag* – This enables the inclusion of the drag term in the axial-induction algorithm.

[21] *TIDrag* – This enables the inclusion of the drag term in the tangential-induction algorithm. If the drag term is included in the tangential-induction algorithm (if *TIDrag* is true) then a singularity exists in the blade element momentum equations.

[22] *TISingularity* – This removes the singularity introduced by *TIDrag* and may improve convergence towards a solution to the blade element momentum equations. Therefore, *TISingularity* is only used if *TIDrag* is set to true.

[23] *DAWT* – This allows WT_Perf to analyze a diffuser augmented water turbine. *This feature is not fully implemented yet and WT_Perf will abort with an error message if this flag is set.*

[24] *Cavitation* – Setting this to true tells WT_Perf to check for cavitation during analysis of a water turbine. There will be a message in the .oup and .bed output files which indicates if cavitation occurs or not. The settings in the following section will be used.

DRAFT

DRAFT

Cavitation Model

WT_Perf predicts that cavitation will occur if the following inequality is true:

$$\sigma + C_{Pmin} < 0,$$

where C_{Pmin} is the minimum pressure coefficient, and σ is the cavitation number defined as

$$\sigma = \frac{PressAtm + Rho * \boldsymbol{g} * \boldsymbol{d} - CavSF * PressVapor}{{}^{1}\!/_{2} * Rho * \boldsymbol{V_{loc}^{2}}}$$

and $\boldsymbol{g}$ is the gravitational accelleration, $\boldsymbol{d}$ is the depth from the free surface of the blade segment being analyzed, and $\boldsymbol{V_{loc}^{2}}$ *is the total induced velocity at that blade element.*

[26] *PressAtm* – The absolute pressure of the atmospheric air. For Standard Temperature and Pressure at sea level, use *101325.0 N/m*^{2} *or 2116.2 lb/ft*^{2}**.**

[27] *PressVapor* – The absolute vapor pressure of the water, and for seawater values are approximately 2300 N/m^{2} or 48 lb/ft^{2}.

[28] *CavSF* – This is a safety factor which is multiplied to the vapor pressure (*PressVapor*), the recommended value is 1.0 or larger.

[29] *WatDepth* – The distance from the water free surface to the seabed (bottom of tower), it is entered in units of meters or feet if using dimensional data, otherwise divide *WatDepth* by the rotor radius.

Turbine Data

[31] *NumBlade* – The number of blades on the turbine. It must be an integer greater than zero.

[32] *RotorRad* – The rotor radius. It is the distance along the preconed blade, and is therefore a number larger than the swept radius if the precone is not zero.

[33] *HubRad* – The hub radius entered in either meters or feet if using dimensional data. Otherwise, divide the hub radius by the rotor radius.

[34] *PreCone* – The precone angle in degrees should be a positive value regardless of whether the turbine is downwind or upwind.

[35] *Tilt* – The tilt angle of the shaft in degrees.

[36] *Yaw* – The yaw angle as specified in degrees.

[37] *HubHt* -- The hub height entered in either meters or feet if using dimensional data. Otherwise, divide the hub height by the rotor radius.

[38] *NumSeg* – This tells WT_Perf how many analysis points there will be along the blade. The input data should be for the centers of the segments.

The next part of this section contains a header (line 39) followed by *NumSeg* lines defining the distributions of *RElm (*the distance along the blade of the center of the segment from the center of rotation), *Twist*, *Chord*, airfoil file number (*AFfile*), and a flag (*PrntElem*) to tell WT_Perf to print output data for that blade element in the .bed file. If you are entering data in nondimensional form, *RElm* and *Chord* must be normalized by *RotorRad*. Enter *Twist* in degrees.

Aerodynamic Data

Since the lines in this section follow the blade segment information, the line numbers listed here should have the value of *NumSeg* above (line 38) added to get the real line number.

[41+] *Rho* – The working fluid density is always entered as a dimensional number. Use either kg/m^{3} or slugs/ft^{3}. For Standard Temperature and Pressure at sea level, use 1.225 kg/m^{3} or 0.00238 slugs/ft^{3} for air. For seawater use 1024 kg/m^{3} or 1.987 slugs/ft^{3}.

[42+] *KinVisc* – For calculating the Reynolds Number, we added the variable *KinVisc*, the kinematic viscosity. For Standard Temperature and Pressure at sea level, use 1.464E-05 m^{2}/sec or 1.576E-04 ft^{2}/sec for air. For seawater use 1.05E-06 m^{2}/sec or 1.13E-05 ft^{2}/sec.

[43+] *ShearExp* – This is the exponent of the power-law wind shear. For the standard 1/7th power law, use 0.143.

[44+] *UseCm* – The airfoil tables are compatible with AeroDyn. Therefore, they may contain pitching moment coefficient and/or minimum pressure coefficient data. If pitching moment coefficient data is contained in the airfoil files, set *UseCm* to true.

[45+] *UseCpmin* – If minimum pressure coefficient data is present in the airfoil files set *UseCpmin* to true. *UseCpmin* must be set to true if checking for cavitation since WT_Perf predicts cavitation based off of the minimum pressure coefficient.

[46+] *NumAF* – This is the number of unique airfoil table files. In the lines that follow, enter the airfoil filenames on separate lines and enclose the strings in quotes or apostrophes (pathnames may be either absolute or relative). The next major section describes the format of these airfoil files.

I/O Settings

The line numbers listed in this section should have both *NumSeg* (line 37) and *NumAF* (line 46) added to them to get the real line number in the input file.

[48+] *UnfPower* – This causes the output files to be written in binary format. It should usually be set to false. The advantage is the files will be smaller and faster to write.

[49+] *TabDel* – When set to true, this flag tells WT_Perf to generate output files with tabs between the columns, instead of using fixed format. Tab-delimited files are best for importing into spreadsheets, while fixed-format files are best for viewing with a text editor or for printing. This has no effect when *UnfPower* is set to true.

[50+] *ConvFlag* – This flag tells WT_Perf how to handle cases that did not converge to within the specified tolerances. The following values are permitted:

0. The non-converged value will be output to the oup and bed files.
1. Values of nines (for example, power is given as 999.999) will be output (*previously this was the OutNines behavior*).
2. *NaN* (not-a-number) will be output instead.

[51+] *Beep* – This enables or disables beeps when an error is encountered.

[52+] *KFact* – When set to true, WT_Perf will output data in the primary results file in “kilo” units. For example, thrust would be in kN or klbf instead of N or lbf, and power would be in kW instead of W.

[53+] *WriteBED* – When this is set to true, WT_Perf will generate a file with the extension .bed containing the blade-element data. Only segments that have their *PrntElem* flag set in the distributed-data block above will be included in the file. The last section of this guide describes the contents of the .bed output file.

[54+] *InputTSR* – When true, WT_Perf will expect the speed data to be tip-speed ratios (TSR) instead of actual wind speeds. This applies to both combined-case and parametric analyses.

[55+] *OutMaxCp* – When this is set to true *and* a parametric analysis is performed, WT_Perf will compute the conditions (wind speed or tip speed ratio, rotor speed, and pitch angle) which resulted in the maximum power coefficient and output these results in the header of the .oup file.

[56+] *SpdUnits* – This string tells WT_Perf what units are used for wind-speed data. Three possible values are valid: "mps" will tell the code that the wind-speed values are in meters/second, "fps" will indicate that they are in feet/second, and "mph" will indicate that they are in miles/hour. If *InputTSR* is true, this parameter is ignored.

Combined-Case Analysis

The line numbers listed in this section should have both *NumSeg* and *NumAF* added to them to get the real line number in the input file.

[57+] *NumCases* – This is the number of combined cases to run. If set to zero, WT_Perf runs no combined cases but performs the old-style parametric analysis.

[58+] The second line is the header for the columns in the combined-cases block. It must not be removed from the file.

In the lines following, enter *NumCases* lines containing a combination of speed (wind speed or TSR), rotor speed (*RotSpd*) in rpm, and *Pitch* in degrees.

If *NumCases* is greater than zero, WT_Perf will do the performance analysis for each case and generate a single table containing wind speed, TSR, rotor speed, pitch, power, torque, thrust, flap moment, power coefficient, and cavitation flag columns.

Parametric Analysis

If the value of *NumCases* above is non-zero, this section is ignored. The line numbers listed in this section should have the values of *NumSeg* and *NumAF* added to them to get the real line number in the input file.

If *NumCases* is zero, WT_Perf varies as many as three parameters in each run: rotor speed in rpm, blade pitch in degrees, and wind speed. Enter the wind speed as a tip-speed ratio or an actual wind speed according to the *InputTSR* flag mentioned above. The first three parameters in this section are *ParRow*, *ParCol*, and *ParTab*. They determine how the output data are tabulated for output. If all three parameters are varied, WT_Perf generates multiple tables of data. Possible values are 1 for rpm, 2 for pitch, and 3 for tip speed ratio or speed (determined by *InputTSR*).

[61+] The *ParRow* parameter determines variation in the table rows.

[62+] The *ParCol* parameter determines column variation in the tables.

[63+] The *ParTab* parameter determines which of the parametric values vary from sheet to sheet (table to table).

The next five parameters in this section tell WT_Perf which of the possible output values should be written to the output file. These can be either true or false.

[64+] *OutPwr* – Output rotor power (kW).

[65+] *OutCp* – Output power coefficient (C_P).

[66+] *OutTrq* – Output rotor torque (N-m or ft-lbf).

[67+] *OutFlp* – Output the flap-bending moment at the hub radius (N-m or ft-lbf).

[68+] *OutThr* – Output the rotor thrust (N or lbf).

[69+] This line tells WT_Perf how to vary the various parameters. The *PitSt*, *PitEnd*, and *PitDel* values define the start, end, and delta pitch angles to use. They are input in degrees. The pitch value is added to the local twist at each segment to determine the angle between the chord line and the plane (or cone) of rotation.

[70+] The *OmgSt*, *OmgEnd*, and *OmgDel* parameters define the start, end, and delta rotor speed in rpm.

[71+] This line specifies the parametric wind speeds. You can either input values in tip-speed ratio (speed of the blade tip divided by the wind speed) or actual wind speeds. If you enable the *InputTSR* flag mentioned above, WT_Perf will expect the following line to be tip-speed ratios. *SpdSt*, *SpdEnd*, and *SpdDel* define the start, end, and delta speed. If *InputTSR* is false, enter actual wind speeds. The *SpdUnits* string mentioned above defines the units for actual wind speeds.

DRAFT

Creating the Airfoil Data Files

The files containing aerodynamic coefficients are compatible with those used by AeroDyn (http://wind.nrel.gov/designcodes/simulators/aerodyn). WT_Perf accepts files formatted to the existing AeroDyn v12 style or the newer, slightly modified style. The differences between the styles are minor. Please see the AeroDyn user's guide for details on the old format of the files.

To tell WT_Perf to assume the file is in the new format, start the first line with the string *AERODYN INPUT FILE* (not case sensitive). The second and third lines are for comments. The fourth line tells WT_Perf how many blocks of data there will be for different Reynolds numbers.

For each block, the first value is the Reynolds number. The second value is the control setting. The next seven lines are for AeroDyn's dynamic-stall model and are ignored by WT_Perf, but you must include them.

A multicolumn table follows. The first column is for the angle of attack in degrees. The angle of attack values must increase monotonically and it is recommended that the values span the range -180 to 180 degrees. The second and third columns are for the lift and drag coefficients. The fourth and fifth columns can be used for the pitching moment (*Cm*) and minimum pressure coefficients (*Cpmin*). If only using one of these coefficients, either *Cm* or *Cpmin*, enter the coefficient data in the fourth column. If using both *Cm* and *Cpmin*, enter *Cm* in the fourth column and *Cpmin* in the fifth column. At the end of each table, add a line containing the string "EOT". The new table format allows one to use different sets of angles of attack for each table. At the very end of the file, after the final "EOT" add a blank line with no text on it.

Output *.bed* file

The .bed output file contains the following variables at each blade segment/sector:

Element: The blade element being analyzed

RElm: The radius of the blade element

IncidAng: The incidence angle (twist + pitch)

Azimuth: The angle which bisects the current rotor sector

Loc Vel: The total induced velocity at the rotor plane

Re: Reynolds number

Loss: The loss factor induced by the hub and/or tip loss models

Axial Ind.: The axial induction factor

Tang. Ind.: The tangential induction factor

Airflow Angle: The angle between the cone of rotation and the total induced velocity vector.

AlfaD: The angle of attack

Cl: The lift coefficient

Cd: The drag coefficient

Cm: The pitching moment coefficient

Cpmin: The minimum pressure coefficient

CavNum: The cavitation number

Cav: Cav will equal T if cavitation is predicted to occur, and F otherwise

Thrust Coef: Thrust coefficient within the sector of the annulus

Torque Coef: Torque coefficient within the sector of the annulus

Power Coef: Power coefficient within the sector of the annulus

Thrust/Len: The thrust within the sector of the annulus divided by the span of the blade element

Torque/Len: The torque produced by the sector of the annulus divided by the span of the blade element

Power: The mechanical power produced by the sector of the annulus multiplied by the number of blades

Converge: This will equal T if solutions were fully converged for all iterations of this case, or F if any iteration (blade element or segment) failed.

Caveats

NREL makes no promises about the usability or accuracy of WT_Perf, which is essentially a beta code. NREL does not have the resources to provide full support for this program. *You may use WT_Perf for evaluation purposes only*.

Acknowledgements

WT_Perf development was funded by the U.S. Department of Energy.

DRAFT

Feedback

If you have problems with WT_Perf, please contact Andy Platt or Marshall Buhl. If either of them have time to respond to your needs, they will do so, but please do not expect an immediate response. Please send your comments or bug reports to:

Andy Platt or Marshall Buhl
NWTC/3811
National Renewable Energy Laboratory
1617 Cole Blvd.
Golden, CO 80401-3393
United States of America

Web: http://wind.nrel.gov
Email: andrew.platt@nrel.gov
Email: marshall.buhl@nrel.gov

References

Buhl, Jr. M.L. “Installing NWTC Design Codes on PCs Running Windows NT®.” National Renewable Energy Laboratory, http://wind.nrel.gov/designcodes/papers/setup.pdf. Last modified Jul. 10, 2006; accessed Nov. 9, 2012. NREL/EP-500-29384. Golden, Colorado.

Buhl, Marshall L. “A New Empirical Relationship between Thrust Coefficient and Induction Factor for the Turbulent Windmill State”, National Renewable Energy Laboratory, http://www.nrel.gov/docs/fy05osti/36834.pdf. Last modified Aug. 2005, accessed Nov. 9, 2012. NREL/TP-500-36834. Golden Colorado.

Hibbs, B.; Radkey, R. L. “Small Wind Energy Conversion Systems (SWECS) Rotor Performance Model Comparison Study” (UNPUBLISHED DRAFT). Pasadena, California: Aerovironment, Inc, November 1981. (http://wind.nrel.gov/designcodes/papers/HibbsRadkey_BEM_Theory.pdf).

Laino, D.J.; Hansen, A.C. “User’s Guide to the Wind Turbine Dynamics Computer Software AeroDyn.” Salt Lake City, Utah: Windward Engineering, LC, August 2001. (http://www.nrel.gov/docs/fy05osti/36881.pdf).

Maniaci, D.C. “An Investigation into WT_Perf Convergence Issues”, 49th AIAA Aerospace Sciences Meeting, AIAA 2011-150, Orlando, Florida, January 2011. (http://wind.nrel.gov/designcodes/papers/AIAA-2011-150-703.pdf).

Wilson, Robert E.; Lissaman, Peter B. S. “Applied Aerodynamics of Wind Power Machines.” Oregon State University, May 1974. (http://ir.library.oregonstate.edu/jspui/bitstream/1957/8140/4/WilsonLissaman_AppAeroOfWindPwrMach_1974.pdf)

Wilson, Robert E.; Lissaman, Peter B. S.; Walker, Stel N. “Aerodynamic Performance of Wind Turbines.” Oregon State University, June 1976. (http://wind.nrel.gov/designcodes/papers/WilsonLissamanWalker_AerodynamicPerformanceOfWindTurbines(1976).pdf).

DRAFT